

**Manuscript version: Published Version**

The version presented in WRAP is the published version (Version of Record).

**Persistent WRAP URL:**

<http://wrap.warwick.ac.uk/116717>

**How to cite:**

The repository item page linked to above, will contain details on accessing citation guidance from the publisher.

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

**Publisher's statement:**

Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk)

Received January 21, 2019, accepted February 8, 2019, date of publication February 21, 2019, date of current version March 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900335

# Convolution-Based Neural Attention With Applications to Sentiment Classification

JIACHEN DU<sup>1</sup>, LIN GUI<sup>2</sup>, YULAN HE<sup>2</sup>, RUIFENG XU<sup>1,3</sup>, (Member, IEEE),  
AND XUAN WANG<sup>1</sup>, (Member, IEEE)

<sup>1</sup>School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518000, China

<sup>2</sup>Department of Computer Science, University of Warwick, Coventry CV47AL, U.K.

<sup>3</sup>Peng Cheng Laboratory, Shenzhen 518000, China

Corresponding author: Ruifeng Xu (xuruifeng@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1636103, Grant 61632011, and Grant 61876053, in part by the Key Technologies Research and Development Program of Shenzhen under Grant JSGG20170817140856618, in part by the Shenzhen Foundational Research Funding under Grant 20170307150024907, and in part by the Innovate U.K. under Grant 103652.

**ABSTRACT** Neural attention mechanism has achieved many successes in various tasks in natural language processing. However, existing neural attention models based on a densely connected network are loosely related to the attention mechanism found in psychology and neuroscience. Motivated by the finding in neuroscience that human possesses the template-searching attention mechanism, we propose to use convolution operation to simulate attentions and give a mathematical explanation of our neural attention model. We then introduce a new network architecture, which combines a recurrent neural network with our convolution-based attention model and further stacks an attention-based neural model to build a hierarchical sentiment classification model. The experimental results show that our proposed models can capture salient parts of the text to improve the performance of sentiment classification at both the sentence level and the document level.

**INDEX TERMS** Natural language processing, sentiment classification, convolutional neural networks, neural attention model.

## I. INTRODUCTION

The task of sentiment classification, which identifies the sentiment polarity of a given sentence or document, has become an attracting topic in Natural Language Processing (NLP) [1]–[3]. Traditional approaches treat this task as text classification, focusing on designing effective features to obtain better performance, such as  $n$ -grams [4], topics extracted by topic models [5] and dependency parse trees [6]. These feature-based methods have achieved some success in sentiment classification, but they are sensitive to the noise in text and require expensive feature engineering.

To solve this problem, Neural Network (NN) based models have been introduced to learn the continuous document representations in an end-to-end manner for sentiment classification. NN based models firstly project the one-hot representation of a word in text into a continuous low-dimensional space [7]–[9], and apply various deep NN

architectures to learn the latent representation of text and perform classification. Two NN architectures are widely used in sentiment classification, namely, Convolutional Neural Networks (CNNs) [10] and Recurrent Neural Networks (RNNs) based on long short-term memory [11]. CNN based neural networks have shown promising performance on various NLP tasks such as sentiment classification [12], [13], question answering [14] and text generation [15]. Although CNN is able to capture salient parts in text, it typically only deals with short-distance relations between words in text and largely ignores long-term dependencies such as words in a considerable distance indicating negations and sentiment transitions. RNN is another promising model commonly used in sentiment classification. For example, Tang *et al.* [16] used gated RNN to model documents for sentiment classification. Different from CNN based model, RNN is better at modeling long-distance semantics in text and capturing contextual information. However, conventional RNN is not capable of focusing on the salient parts in text which are important in sentiment classification.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

More recently, a new direction of deep learning research has emerged, which introduces an attention mechanism to conventional NN-based models. NN with attention is able to attend to specific parts of text as the simulation of human's attention while processing text. The attention mechanism has been widely applied in various downstream applications including caption generation [17], [18], machine translation [19], [20], reading comprehension [21], [22] and text summarization. Although neural attention models give impressive performance in the aforementioned tasks, they are less effective for text classification. For example, Yang *et al.* [23] modified the neural attention mechanism used in the sequence-to-sequence (seq2seq) model and applied it for document-level classification. The attention used in [23] can be seen as a *self-attention* module that takes the hidden states of word-level or sentence-level RNN as input and outputs the attention scores based on the dot product between the hidden states and a global context vector. Compared with non-attention methods, the improvement is marginal in the reported results. The reason is that, in the seq2seq problems, attention is defined based on the alignment between a given text unit (word or sentence) with some observed labels, such as a word in another language (for machine translation) or a Part-Of-Speech tag (for POS tagging). In sentiment classification, there is no target label for each text unit to indicate whether the unit is sentiment-relevant or not. As such, there is no evidence why the densely connected network with one hidden layer attention is efficient for sentiment classification.

Inspired by the cognitive and neuroscience research, we propose a novel neural attention model for sentiment classification. The basic idea is to firstly apply one-dimensional convolution operation to model the information of text unit in its context and capture the attention signals, then use an RNN to autoregressively represent the text sequence with attention signals for sentiment classification. A text unit (words or sentences) with higher attention weight, which conveys more important information, will be highlighted in sentiment classification. Our main contributions in this paper is four-fold:

- we propose a convolution-based model to stimulate human's reading attentions based on cognitive and neuroscience research, justified by mathematical theories.
- We also propose an efficient method for attention signal extraction, which can be applied in a wide range of tasks.
- We propose a new sentence-level sentiment classification architecture named Convolutional-Recurrent Attention Network (CRAN) which combines convolution-based attentions with RNNs. Experimental results show that CRAN can capture salient words in sentence and outperforms several strong baselines.
- Finally, we hierarchically stack CRAN to construct a document-level model named Hierarchical Convolutional-Recurrent Attention Network (*H-CRAN*). Experimental results also show that *H-CRAN* can capture sentence-level attention signals and outperforms

the state-of-the-art models in document-level sentiment classification.

## II. RELATED WORKS

### A. SENTIMENT CLASSIFICATION

Sentiment analysis is one of the main research topics in NLP [2], [3]. It can be carried out at two different levels of granularity: sentence-level and document-level [2]. Generally speaking, sentiment classification can be treated as traditional text classification, and solved by supervised statistical learning models [1]–[4]. Traditional feature-engineering based models usually focus on extracting efficient features including lexical features [4], [24], topic-based features [5], [25] and ontology-based features [26], [27]. With the rapid development of deep learning, NN based models have shown promising performance on sentiment classification. Various architectures of NN including Multi-Layer Perceptron (MLP) [28], CNN [12] and RNN [29] have been proposed to obtain better representations of sentences for classification. In document-level sentiment classification, the hierarchical semantic composition of a document can be modeled by hierarchical models such as Gated Recurrent Neural Network [16] and Hierarchical Attention Network [23].

### B. NEURAL ATTENTION MODELS

Neural Attention models are commonly applied to NLP tasks including neural machine translation [19], text summarization [30] and text inference [22]. The main idea of neural attention is to learn an alignment of a source sequence and a target sequence (sequence-to-sequence tasks) or learn how important a word is based on the matching score with a global context vector (classification tasks). Recent research on neural attention models focuses on the development of methods for computing the alignment scores to generate soft or hard attention signals. Soft attention models represent the alignment score by a probability value between 0 and 1 which is usually computed by a fully-connected neural network [19], [20]. The probability value reflects the importance of an alignment pair between source and target. The advantage of soft attention is that it can be easily trained with other component of the model in an end-to-end way. However, soft attention usually suffers from *Attention Distraction* [18] which refers to the problem of assigning relatively small but nonzero attention value to unrelated parts of a sequence. The *Attention Distraction* problem usually weakens the attention scores assigned to significant parts in text and increases the computation burden.

To alleviate this problem, hard attention models are proposed to force the model to only select important parts and ignore the trivial items. Hard attention assumes that the attention score to be a Boolean value indicating whether selecting the item or not [18]. As in the soft attention model, the hard attention score in the hard attention model is computed by a densely connected network. Since the attention score is a discrete value that cannot be learned by back

propagation, the hard attention model usually relies on optimization methods in reinforcement learning such as policy gradient [31]. The hard attention model is widely applied in NLP tasks like machine reading comprehension [32] and sentiment classification [33]. However, since the weights of models estimated by policy gradient usually have high variance, the hard attention model is extremely hard to be trained and transferred to different data distributions.

Existing neural attention models (hard or soft) usually compute the attention score by a fully connected neural network which only leverages the current source input and ignores the context information. The fully-connected network of the attention model limits the *receptive field* on text sequences, especially when processing long sequences. Moreover, there is few in-depth research on the relation between human's attention mechanism and neural attention models in deep learning. To remedy these problems, we propose a convolution-based attention model which leverage context information in the source sequence to effectively models the human's attention mechanism.

### III. A CONVOLUTION-BASED ATTENTION MODEL

#### A. CONVOLUTION AS A SIMULATION TO HUMAN ATTENTION

While reading text, humans usually pay attention to only a small amount of information presented in visual scenes [34], [35] and only focus on the partial information that is directly related to a task at hand. Cognitive and neuroscience researches have explained this phenomena by many psychological experiments. These experiments show humans depict in brains a cognitive representation or a *search template* of a certain task and try to only focus on text unit which can match the *search template* [36]. Psycho-linguists have proven that template-matching process also helps us concentrate on the important content while processing long texts in our brain [37], [38]. Although this mechanism of attention has been thoroughly investigated in neuroscience and psychology, there is few research on how to leverage these results from psychology and neuroscience into NLP. Motivated by this, we propose a novel model introducing the aforementioned attention mechanism to NLP, using text classification as an example.

Based on an in-depth investigation, we found that *convolution operation* is a natural model to stimulate previously discussed *template-searching* attention mechanism, since the convolution operation is similar to the process of template matching. For textual data, one-dimensional convolution is always applied to the concatenated word vectors or sentence vectors. Without loss of generality, we focus on the word-level attention here. In NN based models, a sequence of text with length  $T$  (padded when necessary) is often represented as

$$x_{0:T-1} = x_0 \oplus x_1 \oplus \dots \oplus x_{T-1}, \quad (1)$$

where  $x_t \in \mathcal{R}^d, t = \{0, 1, \dots, T-1\}$  is the  $d$ -dimensional vector representation of the  $t$ -th word in a text sequence,

and  $\oplus$  is the concatenation operation for vectors. One dimensional convolution applies a filter  $w = w_0 \oplus w_1 \oplus \dots \oplus w_{l-1}$  to a span of  $l$  words in the text sequence to get a convolutional similarity score  $c_t$ . The convolution operation applies sequential linear transformation to each continuous subsequence of length  $l$  in  $[x_0, x_1, \dots, x_{T-1}]$  by:

$$c_t = f(< w, x_{t:t+l-1} > + b) \quad (2)$$

$$= f\left(\|w\| \times \|x_{t:t+l-1}\| \times \frac{\langle w, x_{t:t+l-1} \rangle}{\|w\| \times \|x_{t:t+l-1}\|} + b\right). \quad (3)$$

In Equation 2, the subsequence of text is concatenation of word vectors  $x_{t:t+l-1} = x_t \oplus x_{t+1} \oplus \dots \oplus x_{t+l-1}$ ,  $\langle \cdot, \cdot \rangle$  is dot product of two vectors as  $\langle a, b \rangle = a^T b$ ,  $\|\cdot\|$  is the second-order norm  $\mathcal{L}^2$  of vectors,  $f$  is the non-linear transformation (i.e. sigmoid, hyperbolic tangent or ReLU function). Note that  $w$  and  $x_{t:t+l-1}$  are  $l \times d$ -dimensional vectors, assuming that each dimension has its own distribution. According to the Chebyshev Law, for any  $w$  and any  $x_{t:t+l-1}$ , if  $l \times d$  is large enough, then for any  $\varepsilon > 0$ , there exists  $M$  such that  $P(|M - \|w\| \times \|x_{t:t+l-1}\|| < \varepsilon) = 1$ . The shape of convolution filters  $l \times d$  is usually larger than 25, which satisfies the assumption of Chebyshev Law. As a result, we can replace  $\|w\| \times \|x_{t:t+l-1}\|$  in equation (3) by  $M$ . If we define a function  $F(x) = f(Mx)$  to replace the original function  $f$  and replace  $b'$  by  $b' = b/M$ , we obtain:

$$c_{t:t+l-1} = F(\cos(w, x_{t:t+l-1}) + b') \quad (4)$$

In Equation 4, it is noticed that  $F$  is only a transformation function that satisfies  $F'(x) > 0$ . The convolutional filter  $w$  can be regarded as a *search template* in human's attention while reading text as discussed previously. And  $c_{t:t+l-1}$  can be treated as the cosine similarity between the search template and the part of text which is currently processed.  $b'$  in equation 4 is the threshold of this similarity. When the similarity is greater than  $b'$ , the textual part being processed can be seen as task-relevant; otherwise it is task-irrelevant. Thus we show that one-dimension convolution is precisely the process of calculating the similarity between text and the *attentive-search templates*, which can be seen as a simulation to human reading attention.

#### B. CONVOLUTION-BASED ATTENTION MODEL

As we showed in Section III, the output of one-dimensional convolution operation can be seen as an attention signal on the text sequence. Motivated by this finding, we propose an convolution-based attention model to better capture the important parts in text sequence for sentiment classification. In order to reduce the variance in attention signals obtained by convolution, we apply multiple convolutional filters to the vector representation of a text sequence and average the outputs of all the filters to get a smoothed attention signal. The convolution filters are denoted by  $[w^1, w^2, \dots, w^m]$  ( $m$  is the number of convolutional filters), and the corresponding attentional signals are  $[c^1, c^2, \dots, c^m]$ . After averaging the attentional signals along the filter-axis, we can obtain the smooth attention signal,  $c_{t,t+l-1} \in \mathcal{R}$ , representing the

importance of a word sequence starting from word  $t$  with length  $l$ .

$$c_{t,t+l-1} = \sum_{i=1}^m c_{t,t+l-1}^i = \sum_{i=1}^m f(\langle w^i, x_{t:t+l-1} \rangle + b) \quad (5)$$

In order to disentangle the attention signal for every single word, we average attention signals involving word  $t$ . Equation (4) shows that  $[c_{t-l+1,t}, \dots, c_{t:t+l-1}]$  involves word  $t$ . The attention signal  $a_t$  for word  $t$  can be written as

$$c'_t = \frac{1}{l} \sum_{t'=t-l+1}^t c_{t',t'+l-1} \quad (6)$$

Furthermore, to normalize the attention signals, we apply softmax function to  $[c'_0, \dots, c'_{T-1}]$ :

$$a_t = \frac{\exp c'_t}{\sum_{t'} \exp c'_{t'}} \quad (7)$$

$a_t \in \mathcal{R}$  is the final output of our attention model and represents the importance of word  $t$  in the whole sentence. The proposed convolution-based attention model is shown in Figure 1.

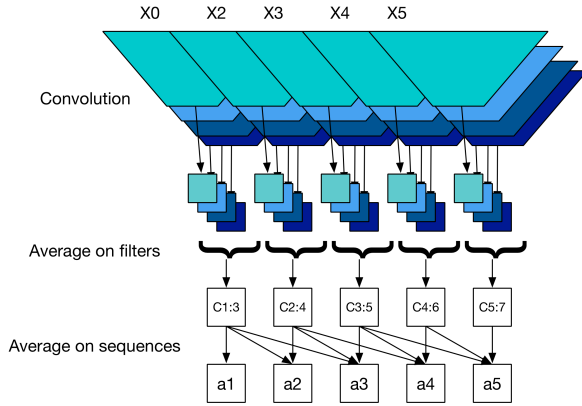


FIGURE 1. Convolution-based model of attention signal extraction.

## IV. PROPOSED MODELS

### A. CONVOLUTIONAL-RECURRENT ATTENTION NEURAL NETWORKS (CRAN)

Based on the attention model we described in Section III, we propose a model named Convolutional-Recurrent Attention Network (CRAN) that combines RNN with the convolutional attention model. The reason we use RNN as the word encoder rather than directly applying the conventional CNN architecture [12] is that the conventional CNN uses a pooling operation to aggregate the convolutional results of the whole sentence, which ignore some global semantics and long-distance dependencies in text. On the other hand, RNN with gate mechanism such as LSTM and GRU, is designed for handling the long-distance dependencies. We speculate

that combining RNN with our proposed CNN-based attention model will give better performance compared with conventional CNN, and our experimental results which will be presented in Section V-D and 2 confirm our hypothesis.

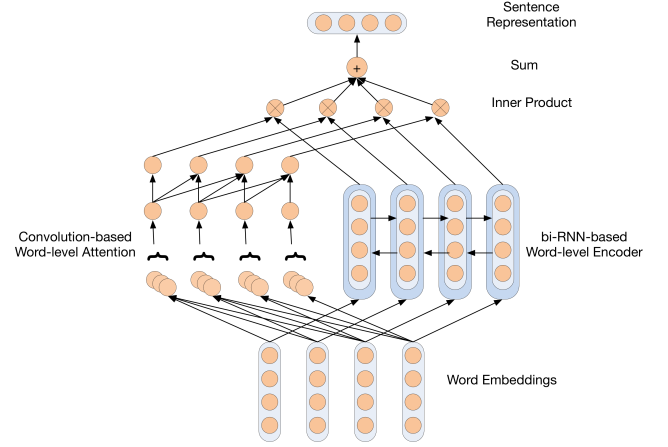


FIGURE 2. Architecture of the convolutional-recurrent attention network (CRAN).

The overall architecture of the CRAN is shown in Figure 2. It consists of two main parts: an RNN as the text encoder and a CNN as the attention extractor. We describe the details of these two parts in the following subsections.

#### 1) RNN-BASED WORD-LEVEL ENCODER

To better model the semantic information of text, we use bidirectional LSTM [11] to derive the hidden state of each word by summarizing the information from both forward and backward directions. Forward LSTM and backward LSTM are denoted as  $\overrightarrow{LSTM}$  and  $\overleftarrow{LSTM}$ , whereas  $\overrightarrow{LSTM}$  reads words from left to right and  $\overleftarrow{LSTM}$  in reverse direction,

$$\overrightarrow{h}_t = \overrightarrow{LSTM}(x_t, \overrightarrow{h}_{t-1}), \quad t \in [0, T-1] \quad (8)$$

$$\overleftarrow{h}_t = \overleftarrow{LSTM}(x_t, \overleftarrow{h}_{t+1}), \quad t \in [0, T-1] \quad (9)$$

We get the representation of each  $x_t$  by concatenating the forward hidden state  $\overrightarrow{h}_t$  and the backward hidden state  $\overleftarrow{h}_t$ , i.e.,  $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$

#### 2) CONVOLUTION-BASED WORD-LEVEL ATTENTION

As discussed in Section III, we use a convolution operation to model the attention signals in sentences. Suppose the text sequence is  $[x_0, x_1, \dots, x_{T-1}]$ , the corresponding attention signals extracted by our model is  $[a_0, \dots, a_{T-1}]$ . Note that the attention signals are all scalar values in the range of 0 to 1:

$$c_{t,t+l-1} = \sum_{i=1}^m \text{Conv}(x_{t:t+l-1}) \quad (10)$$

$$c'_t = \frac{1}{l} \sum_{t'=t-l+1}^t c_{t',t'+l-1} \quad (11)$$

$$a_t = \frac{\exp c'_t}{\sum_{t'} \exp c'_{t'}} \quad (12)$$



Here,  $Conv$  is the one-dimensional convolution operation,  $a_t$  is the attention signal assigned to word  $x_t$ .

### 3) SENTENCE REPRESENTATION

We use the product of attention signal  $a_t$  and the corresponding hidden state vector of RNN,  $h_t$ , to represent word  $t$  in text. The representation of the whole sequence of words is obtained by averaging the word representations:

$$r = \frac{1}{T} \sum_{t=0}^{T-1} a_t h_t \quad (13)$$

Then the representation of sentence is fed to a one-layer fully connected network by

$$p = \text{softmax}(W_c s + b_c) \quad (14)$$

where  $p$  is the predicted probability of sentiment label,  $W_c$  and  $b_c$  are parameters of the classification layer.

## B. HIERARCHICAL CONVOLUTIONAL-RECURRENT ATTENTION NEURAL NETWORKS (H-CRAN)

For document-level sentiment classification, we construct a hierarchical classification model based on CRAN in Section IV-A. Assuming that a document has  $D$  sentences,  $[s_0, \dots, s_{D-1}]$ , and sentence  $s_d$ ,  $d \in [0, D-1]$ , with length  $T$  is denoted as  $[x_0^d, \dots, x_T^d]$ . As we showed in Equation (13) in Section IV-A, we can use CRAN to obtain the representation with attention  $a_d$  of sentence  $d$ , denoted as  $r_d$ . The representations  $[r_0, \dots, r_{D-1}]$  of all sentences in a document can be computed through the same CRAN, which we called the word-level CRAN. We then apply a sentence-level CRAN to the representations of sentences to get the document-level representation vector. We call this model *Hierarchical Convolutional-Recurrent Attention Neural Networks (H-CRAN)*. The whole architecture of H-CRAN is shown in Figure 3.

The sentence-level CRAN has the same structure as the word-level CRAN. The only difference is that the input to the *sentence-level* CRAN is vectors of sentences instead of words in a document. The sentence-level CRAN firstly encodes the output of the word-level CRAN by a sentence-level bidirectional LSTM:

$$\overrightarrow{h_d^{sent}} = \overrightarrow{LSTM}(r_d, \overrightarrow{h_{d-1}^{sent}}), \quad t \in [0, D-1] \quad (15)$$

$$\overleftarrow{h_d^{sent}} = \overleftarrow{LSTM}(r_d, \overleftarrow{h_{d+1}^{sent}}), \quad t \in [D-1, 0] \quad (16)$$

$$h_d^{sent} = [\overrightarrow{h_d^{sent}}, \overleftarrow{h_d^{sent}}] \quad (17)$$

$\overrightarrow{LSTM}$ ,  $\overleftarrow{LSTM}$  are the forward and backward sentence-level LSTM, respectively,  $\overrightarrow{h_d^{sent}}$ ,  $\overleftarrow{h_d^{sent}}$  are the corresponding hidden states of LSTM of both directions,  $h_d^{sent}$  as concatenation of the two hidden states is the vector representation of sentence  $d$  in a document.

Following the same procedure of extracting attention signals at the word-level as presented in Section IV-A.2,

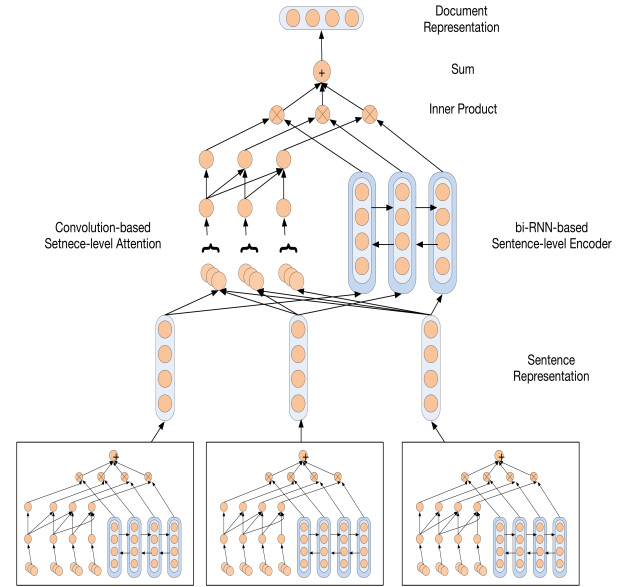


FIGURE 3. Architecture of the hierarchical convolutional-recurrent attentional network (H-CRAN).

we apply  $m$  convolution filters to extract the attention signal for each sentence in a document by:

$$c_{d,d+l-1}^{sent} = \sum_{i=1}^m Conv(r_{d:d+l-1}), \quad (18)$$

$$c_d^{sent'} = \frac{1}{l} \sum_{d'=d-l+1}^d c_{d',d'+l-1}, \quad (19)$$

$$a_d^{sent} = \frac{\exp c_d^{sent'}}{\sum_{d'} \exp c_{d'}^{sent'}}. \quad (20)$$

Here,  $a_d^{sent}$  is the sentence-level attention which shows the significant of a sentence in the whole document.

A given document can be represented as the average of products of the sentence-level RNN's hidden states and the corresponding sentence-level attentions:

$$r^{doc} = \frac{1}{D} \sum_{d=0}^{D-1} a_d^{sent} h_d^{sent}, \quad (21)$$

Finally,  $r^{doc}$  is fed to a softmax classifier to obtain the probability distributions of classes.

## C. MODEL TRAINING

We use the cross-entropy between the predicted class probabilities and the ground-truth labels as the loss function of our model. All components can be trained end-to-end by minimizing the loss function,

$$\mathcal{L} = - \sum_i \sum_j y_i^j \log p(\hat{y}_i^j | x_i, z_i) + \lambda \|\theta\|^2 \quad (22)$$

where  $i$  is the index of data and  $j$  is the index of class.  $\lambda \|\theta\|^2$  is the  $L_2$ -regularization term and  $\theta$  is the parameter set of

our model. We use Adam algorithm [39] to train the proposed model with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$  and  $\epsilon = 1e-9$ . The learning rate is varied over the training procedure, which is inspired by the training strategy of Transformer [40]. The learning rate is set by

$$lr = \min(step^{-\frac{1}{2}}, step \times pre\_step^{-\frac{3}{2}}) \quad (23)$$

where  $lr$  is the learning rate,  $step$  is the number of training steps, and  $pre\_step$  is the number of steps in pre-training phase (warmup training). In our experiments, we set  $pre\_step = 1000$  for sentence-level classification and  $pre\_step = 3000$  for document-level classification.

## V. EXPERIMENTS ON SENTENCE-LEVEL SENTIMENT CLASSIFICATION

In this section, we investigate the empirical performance of our proposed CRAN on various sentence-level sentiment classification datasets and compare it with the state-of-the-art models for sentiment classification.

### A. DATASETS

We choose four sentence-level sentiment classification datasets to evaluate our proposed model.

- **MR**: Binary sentiment classification dataset including 10,662 movie reviews. [41].
- **SST-1**: 5-classes sentence-level sentiment dataset of 11,855 sentences [42].
- **SST-2**: Simplified version of SST-1, with neural classes removed and only containing binary labels (positive and negative), the size is 9,613.
- **Subj**: Subjectivity dataset of 10,000 user-generated reviews with binary categories [43].

Each of the datasets contains about 10k sentences. We conduct experiments on the standard test sets on SST dataset. For the remaining datasets without standard train/test split, we use 10-fold cross validation.

### B. MODEL TRAINING AND HYPER-PARAMETERS

Inspired by the pre-train strategy proposed in [44], we use two different initialization methods for convolutional attention layer:

- **CRAN<sup>rand</sup>**: The weights are initialized by a uniform distribution in the range of  $(-0.01, 0.01)$ .
- **CRAN<sup>pretrain</sup>**: We first pre-trained a one-dimensional CNN classifier with filter-axis pooling proposed in III to obtain the initial weight of the attention layer. The weights of pre-trained CNN are set as initial values of the convolutional attentional layer in CRAN.

In all experiments, the word embeddings are initialized by the 300-dimensional word2vec vectors trained on the Google News data [8]. The number of hidden state of LSTM and convolutional filters is set to 150, the width of convolution filter is set to 3, dropout rate during training is 0.3. All hyper-parameters are tuned to obtain the best performance

based on 5-fold cross validation on the training set for each dataset.

### C. BASELINES

We use several state-of-the-art models as baselines for comparing with our proposed model:

- **NBOW**: Text classifier which sums the word vectors within a sentence and applies a softmax classifier.
- **CNN** [12]: One-dimensional CNN with max pooling, the filter size is 3,4 and 5.
- **LSTM** [29]: The last hidden state of LSTM is fed to a linear sentiment classifier with softmax function.
- **LSTM+attention** [23]: The attention-based LSTM model proposed by [23]. Since datasets used in our experiments are for sentence-level classification, we implement a flatten variant of this model without aggregating the attention signals from sentences to form the document-level attention signal.
- **Tree-LSTM** [45]: Tree-Structured LSTM networks for sentence classification.
- **Multi-Task** [46]: Shared-layer multi-task learning model trained on SST-1, SST-2 and Subj datasets.
- **Sent-Type CNN** [47]: Sentence-level sentiment classifier which leverages several distinct CNN classifiers according to the sentence types.

### D. RESULTS AND ANALYSIS

The experimental results for sentence-level sentiment classification are listed in Table 1. When comparing the performance of the two variants of our model, it is observed that CRAN<sup>rand</sup> performs better than CRAN<sup>pretrain</sup>. One possible reason is that the weights of the convolutional attention layer of CRAN<sup>pretrain</sup> are firstly trained by using different network architecture. The initial weights obtained by pre-training may lead to incompatible results during the fine-tuning phase.

**TABLE 1. Results of our proposed CRAN against baselines. Results marked \* are models that need external tools or resources.**

Model	MR	SST-1	SST-2	Subj
NBOW	77.1	42.1	79.0	90.8
CNN	81.5	48.0	88.1	93.4
LSTM	80.1	46.2	85.2	91.2
LSTM+attention	82.0	48.0	86.1	93.2
Tree-LSTM*	-	<b>50.6</b>	86.9	-
Multi-Task*	-	49.6	87.9	<b>94.1</b>
Sent-Type CNN	82.3	48.5	88.3	-
CRAN <sub>rand</sub>	<b>83.8</b>	50.1	<b>88.9</b>	<b>94.1</b>
CRAN <sub>pretrain</sub>	81.8	49.0	87.8	94.0

Compared with other state-of-the-art models in sentiment classification, CRAN gives the best performance on three out of four datasets. The convolution-based attention extractor of CRAN is similar to the traditional CNN. However, CRAN combines the merits of RNN and CNN to better model sentences. Experimental results show that CRAN improves upon the traditional CNN by 1% and outperforms LSTM by 3% on average. CRAN also gives superior results than LSTM

with attention. This shows that our proposed attention modelling method can capture the class-relevant information from text more accurately. Tree-LSTM outperforms our model on SST-1 by 0.4%. However, it needs an external parser to derive the tree-structure of each sentence, and the results listed in Table 1 is obtained on the exact parsing results of sentences labelled by annotators. It is worth noting that our models are comparable with RNN with multi-task learning [46]. This model is an extremely strong baseline which was trained jointly on four datasets.

## VI. EXPERIMENTS ON DOCUMENT-LEVEL SENTIMENT CLASSIFICATION

To validate the efficiency of our proposed Hierarchical Convolutional-Recurrent Neural Network (H-CRAN), we conduct experiments on large-scale document-level datasets.

### A. DATASETS

For document-level experiments, we use four most commonly used datasets. Yelp 2013/2014/2015 are restaurant reviews labeled with 5-star ratings used in the Yelp Challenges [48]. IMDB is a dataset of movie review in which documents are manually labeled with 10-class ratings [49]<sup>1</sup>.

### B. MODEL TRAINING AND HYPER-PARAMETERS

According to the results reported in Section V-D, CRAN\_rand outperforms CRAN\_pretrain. Therefore, CRAN\_rand is used for both word-level and sentence-level models in H-CRAN. Parameters used in document-level experiments are set as follows. The number of hidden units of word-level LSTM and convolution filters is 300, the length of convolution filter is 3, the number of hidden units of sentence-level LSTM and convolution filters is 150, the dropout rate is 0.3 and mini-batch size is 128.

### C. BASELINES

We compare H-CRAN with the following baselines:

- **NBOW**: The NBOW sums the word vectors within the document and applies a softmax classifier.
- **Paragraph Vector** [50]: Paragraph Vector used for sentiment classification.
- **CNN** [12]: Convolutional neural networks with max pooling applied to the whole document.
- **Conv-GRNN** [16]: Hierarchical document classification model which composes a CNN-based word encoder and a GRU-base sentence encoder.
- **LSTM-GRNN** [16]: Same as **Conv-GRNN** but with LSTM as the word encoder.
- **HAN** [23]: Hierarchical Attention Networks (HAN) for document classification which uses a fully-connect neural network as an attention extractor.

- **Text Concept Vector** [28]: Neural network that Leverages word embeddings combined with concepts extracted from a knowledge base.

**TABLE 2.** Classification accuracy of our proposed CRAN against baselines.

Model	Yelp 2013	Yelp 2014	Yelp 2015	IMDB
NBOW	56.8	57.5	55.4	31.6
Paragraph Vector	57.7	59.2	60.5	34.1
CNN	62.7	61.4	64.5	40.6
Conv-GRNN	63.7	65.5	66.0	42.5
LSTM-GRNN	65.1	67.1	67.6	45.3
HAN	68.2	70.5	71.0	49.4
Text Concept Vector	67.8	69.2	71.5	<b>50.5</b>
H-CRAN	<b>68.7</b>	<b>71.5</b>	<b>73.0</b>	<b>50.2</b>

### D. RESULTS AND ANALYSIS

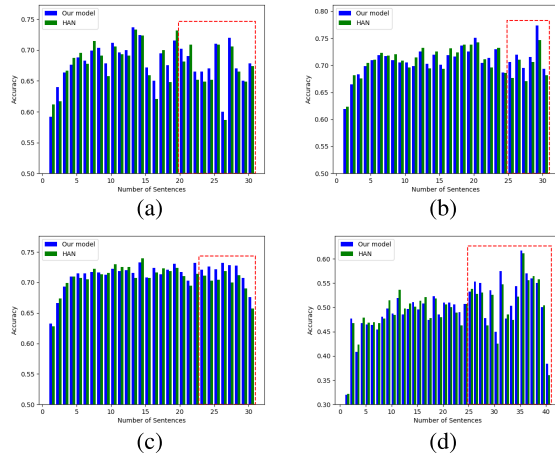
The experimental results of document classification are shown in Table 2. Firstly, we observe that NBOW, Paragraph Vector and CNN perform badly on document-level sentiment classification. The reason is that these models treat the whole document as a long sequence of words and ignore the latent semantic structures of documents. By leveraging the semantic structures of document, Conv-GRNN and LSTM-GRNN use stacked word-level and sentence-level encoders to model the whole document. HAN and our model H-CRAN use a similar hierarchical architecture as that in LSTM-GRNN and Conv-GRNN, but with an additional attention mechanism to extract salient words in sentences and salient sentences in the document. As we can see in Table 2, introducing attention extractor greatly improves the classification accuracy by more than 3.5%. Furthermore, when compared the previously proposed attention-base document model HAN with our model H-CRAN, we observe an improvement of 1.5% on average. The improvement is more prominent on Yelp 2015 in which H-CRAN outperforms HAN by 2%.

### E. IMPACT OF DOCUMENT LENGTH

To investigate the difference in performance of convolution-based attention over documents with different number of sentences, we compare the performance of our model and HAN [23]. Figure 4 shows the classification results of both models on the four datasets. It is observed that both models achieve high accuracies on documents with moderate lengths, but perform relatively worse on documents which are too long or too short. The reason is that documents with fewer sentences usually contain less meaningful information and too long documents contain more redundant words and hence tend to be more noisy. Comparing our model with HAN, we observe that both models perform comparably on documents with a smaller number of sentences (less than 20). However, it is interesting to observe that, when the number of sentences in a document exceeds 20, our model outperforms HAN significantly. The reason is that the convolution operation used as attention extractor in our model

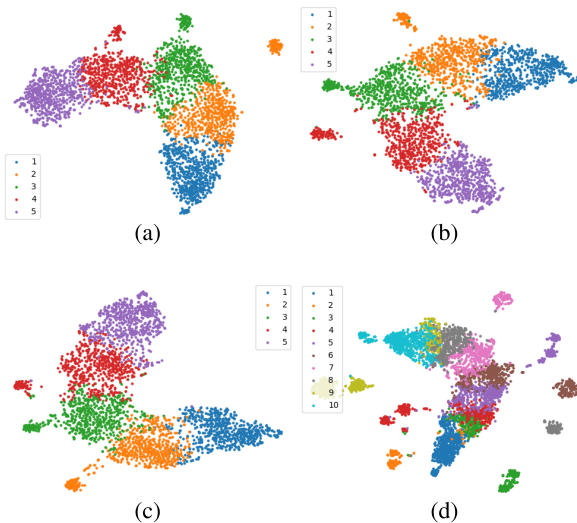
<sup>1</sup>The size of Yelp data sets are separately 211,245, 476,191, 612,636, the size of IMDB is 115,831





**FIGURE 4.** Classification accuracy of our proposed model and HAN on four document-level sentiment classification datasets. (a) Yelp-2013. (b) Yelp-2014. (c) Yelp-2015. (d) IMDB.

has a wider *receptive field* on the text sequence. Therefore convolution-based attention model performs better on longer text sequences which potentially contain more long-term dependencies.

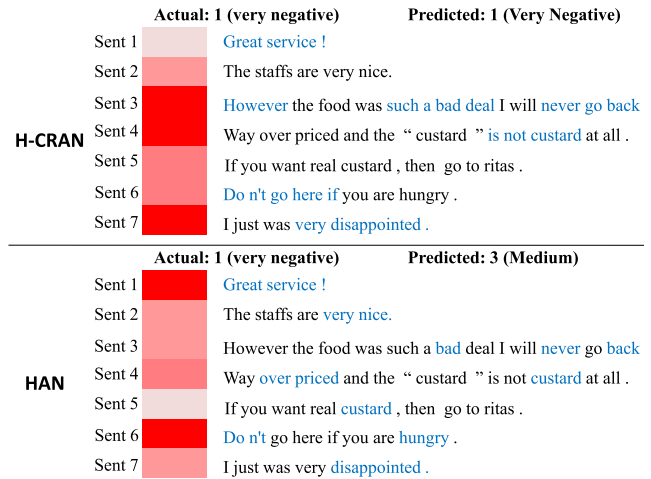


**FIGURE 5.** Visualization of document representations learned by our model on four document-level sentiment classification datasets. (a) Yelp-2013. (b) Yelp-2014. (c) Yelp-2015. (d) IMDB.

## F. VISUALIZATION OF DOCUMENT REPRESENTATIONS

To visualize the representations of documents learned by our proposed model, we extract the intermediate vector representations, feed them to the final classifier and apply t-SNE [51] to project the high-dimensional representations to the two-dimensional space. The t-SNE vectors of document representations are shown in Figure 5. Each point in the figure represents a document and different color of points indicates the ground-truth sentiment class of a document. It is observed that the learned document representations

of different sentiment classes are well-separated in the two-dimensional space. We can thus conclude that our model is able to capture the sentiment information inherent in the documents effectively.



**FIGURE 6.** Visualization of attention signals derived from an example review in the Yelp 2015 dataset.

## G. CASE STUDY AND ERROR ANALYSIS

In order to show that our attention-base model H-CRAN can better extract both sentence-level and word-level attention signals, we visualize the attention values of H-CRAN and the Hierarchical Attention Network (HAN) [23] in Figure 6. The heatmap on the left shows the sentence-level attention signals which indicate the contribution of each sentence towards the overall document-level sentiment classification and words in blue show the salient parts in each sentence. We select a document from the Yelp 2015 which expresses very negative sentiment towards a restaurant. In this example, the first two sentences express appreciation to the service. But the word ‘However’ in Sentence 3 indicates a change in attitude and the remaining sentences express more negative sentiment. To correctly predict the sentiment of this review, the model needs to recognise sentiment transition signified by the word ‘However’ in Sentence 3, and focus on the second parts of this document. The sentence attention signal extracted by HAN showed that HAN failed to identify the sentiment transition and outputs the wrong sentiment label. On the contrary, H-CRAN captures the sentiment transition of this document successfully by assigning a high attention value to Sentence 3, which is crucial for document-level sentiment classification. For the word-level attention, we found that both H-CRAN and HAN are able to capture words expressing strong sentiment in text with some difference. HAN tends to capture salient words in isolation in sentences, but our model is capable of finding more multi-word expressions, such as *such a bad deal* and *very disappointed*. The results of visualization show that our proposed convolutional attention can capture broader context in documents compared to traditional attention models, which

partly explain the better performance of H-CRAN over HAN in document classification.

## VII. CONCLUSION

In this paper, we have shown that the convolution operation is a feasible and effective mechanism for extracting attentions from text sequences. Based on this finding, we have proposed a novel attention extraction model based on the convolution operation. Utilizing this convolution-based attention model, we have introduced a new neural network architecture which combines RNN with our attention model, and further proposed a hierarchical variant for document-level sentiment classification. We have conducted extensive experiments on both sentence-level and document-level datasets and observe from the experimental results that: (1) our model is capable of extracting salient parts from sentences and documents; (2) our model can combine the merits of CNN and RNN to improve the sentiment classification performance; (3) the visualization of attentions extracted by the model shows its impressive capability to capture sentiment transitions in discourses. In future works, we will extend the proposed convolution-based attention model to other tasks such as text generation and sequence to sequence learning.

## REFERENCES

- [1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, vol. 10, 2002, pp. 79–86.
- [2] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.
- [3] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 74–80, Nov./Dec. 2017.
- [4] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2012, pp. 90–94.
- [5] S. Zelikovitz and H. Hirsh, "Using LSI for text classification in the presence of background text," in *Proc. 10th Int. Conf. Inf. Knowl. Manage.*, 2001, pp. 113–118.
- [6] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 786–794.
- [7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
- [8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, (2013). "Efficient estimation of word representations in vector space." [Online]. Available: <https://arxiv.org/abs/1301.3781>
- [9] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 427–431.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [12] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.
- [13] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 655–665.
- [14] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.
- [15] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 627–637.
- [16] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1422–1432.
- [17] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.
- [18] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
- [19] D. Bahdanau, K. Cho, and Y. Bengio, (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: <https://arxiv.org/abs/1409.0473>
- [20] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
- [21] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, "Attention-over-attention neural networks for reading comprehension," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 593–602.
- [22] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov, "Gated-attention readers for text comprehension," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1832–1846.
- [23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.
- [24] D. Bespalov, B. Bai, Y. Qi, and A. Shokoufandeh, "Sentiment classification based on supervised latent n-gram analysis," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 375–382.
- [25] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 375–384.
- [26] Y. Dang, Y. Zhang, and H. Chen, "A lexicon-enhanced method for sentiment classification: An experiment on online product reviews," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 46–53, Jul./Aug. 2010.
- [27] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis," *Knowl.-Based Syst.*, vol. 69, pp. 45–63, Oct. 2014.
- [28] Y. Li *et al.*, "Incorporating knowledge into neural network for text representation," *Expert Syst. Appl.*, vol. 96, pp. 103–114, Apr. 2018.
- [29] A. Graves, (2013). "Generating sequences with recurrent neural networks." [Online]. Available: <https://arxiv.org/abs/1308.0850>
- [30] A. M. Rush, S. Chopra, and J. Weston, (2015). "A neural attention model for abstractive sentence summarization." [Online]. Available: <https://arxiv.org/abs/1509.00685>
- [31] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.
- [32] M. Hu, Y. Peng, Z. Huang, X. Qiu, F. Wei, and M. Zhou, (2017). "Reinforced mnemonic reader for machine reading comprehension." [Online]. Available: <https://arxiv.org/abs/1705.02798>
- [33] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang, "Aspect-level sentiment classification with heat (hierarchical attention) network," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 97–106.
- [34] X. You, Q. Li, D. Tao, W. Ou, and M. Gong, "Local metric learning for exemplar-based object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1265–1276, Aug. 2014.
- [35] Z. Zhu *et al.*, "An adaptive hybrid pattern for noise-robust texture analysis," *Pattern Recognit.*, vol. 48, no. 8, pp. 2592–2608, 2015.
- [36] J. Duncan and G. W. Humphreys, "Visual search and stimulus similarity," *Psychol. Rev.*, vol. 96, no. 3, p. 433, 1989.
- [37] G. McKoon and R. Ratcliff, "Inference during reading," *Psychol. Rev.*, vol. 99, no. 3, p. 440, 1992.
- [38] M. J. Green and D. C. Mitchell, "Absence of real evidence against competition during syntactic ambiguity resolution," *J. Memory Lang.*, vol. 55, no. 1, pp. 1–17, 2006.
- [39] D. P. Kingma and J. Ba, (2014). "Adam: A method for stochastic optimization." [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [40] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [41] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 115–124.

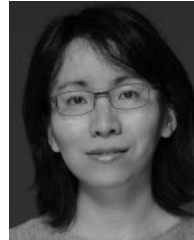
- [42] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.
- [43] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, p. 271.
- [44] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3079–3087.
- [45] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics*, 2015, pp. 1556–1566.
- [46] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2873–2879.
- [47] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.
- [48] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, and S. Ma, "Do users rate or review?: Boost phrase-level sentiment labeling with review-level sentiment classification," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 1027–1030.
- [49] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 193–202.
- [50] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
- [51] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



**JIACHEN DU** is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include sentiment analysis, dialogue generation, and deep learning.



**LIN GUI** received the Ph.D. degree from the Harbin Institute of Technology, Shenzhen, China. He is a Marie Skłodowska-Curie Individual Fellow with the University of Warwick, U.K. His research interests include machine learning, natural language processing, and sentiment analysis.



**YULAN HE** received the Ph.D. degree in spoken language understanding from the University of Cambridge, U.K. She is currently a Professor of computer science with the University of Warwick, U.K. She has published over 150 papers in the areas of text and data mining, sentiment analysis and opinion mining, social media analysis, recommender systems, learning analytics, and spoken dialogue systems.



**RUIFENG XU** received the Ph.D. degree in computer science from The Hong Kong Polytechnic University. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He has published more than 100 papers in natural language processing, sentiment analysis, and social media analysis.



**XUAN WANG** received the Ph.D. degree in computer science from the Harbin Institute of Technology, Shenzhen, China, where he is currently a Professor with the School of Computer Science and Technology. His research interests include cybernetic security, natural language processing, and artificial intelligence.

...